Demo: from running a test to displaying the results

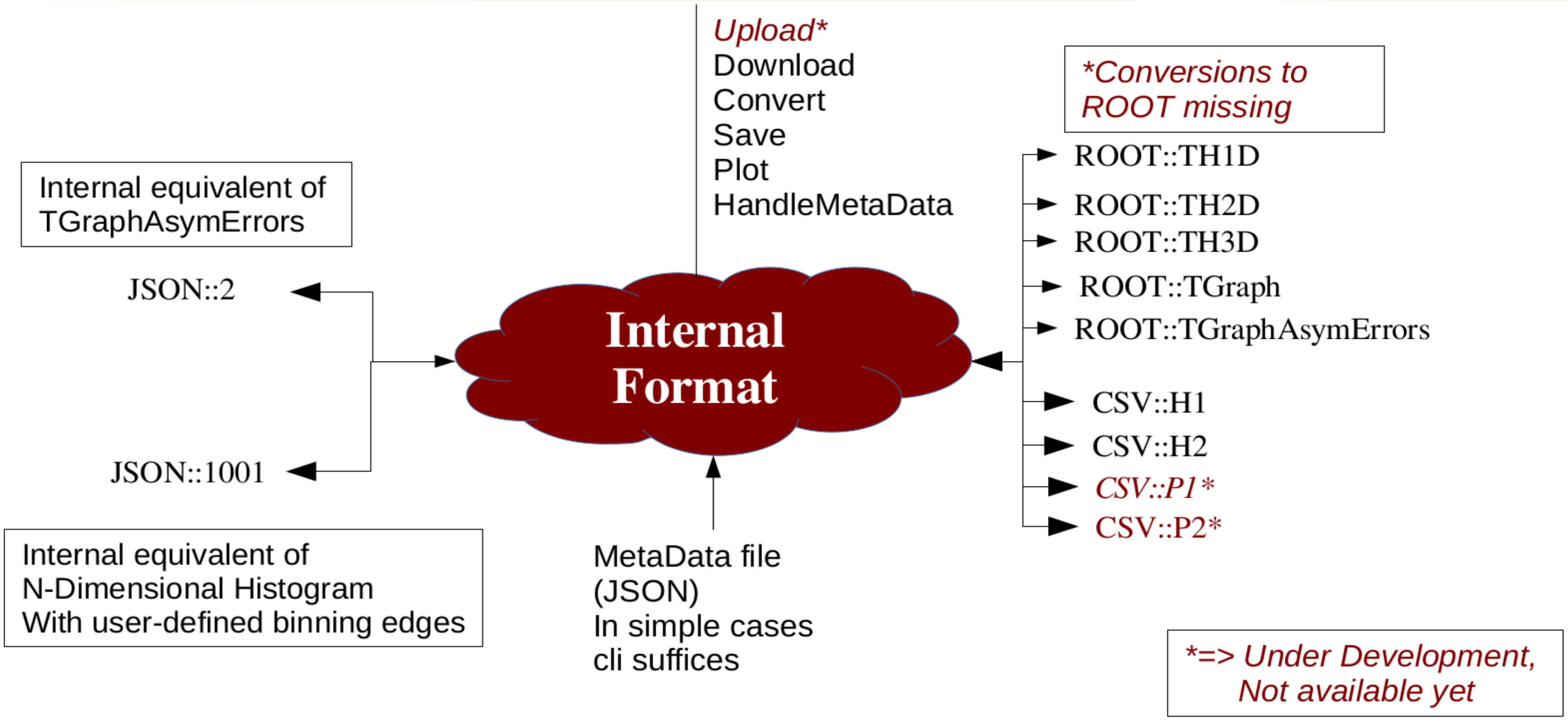
Outline:

- The conversion tool.
- Prerequisits.
- Providing the metadata.
- Extracting histogram or TGraph from a root file and creating json file for upload to DoSSiER
- Uploading the data using the Web Application.
- Displaying the result.

Ancillary Tools: Conversion Tool

Python program

- to read and convert histograms from/to different formats: ROOT, ASCII (CSV), JSON
- Download from and upload to DoSSiER
- Inspect and interact with histograms (*matplotlib*)
- CLI (script integration) or API (integration in python programs) are available



*Upload**
Download
Convert
Save
Plot
HandleMetaData

**Conversions to ROOT missing*

Internal equivalent of TGraphAsymErrors

JSON::2

JSON::1001

Internal Format

ROOT::TH1D
ROOT::TH2D
ROOT::TH3D
ROOT::TGraph
ROOT::TGraphAsymErrors

CSV::H1
CSV::H2
*CSV::P1**
*CSV::P2**

Internal equivalent of
N-Dimensional Histogram
With user-defined binning edges

MetaData file (JSON)
In simple cases cli suffices

**=> Under Development, Not available yet*

For more details see Andrea's talk.

Prerequisits (Linux)

1) You need python, various python modules(e.g. numpy) and Root with python support. If it doesn't exist on your System get the 2.7 version of Anaconda from:
https://www.continuum.io/downloads
2) then build root against the python version provided by Anaconda.
3) Finally obtain the upload tool from gitlab:
https://gitlab.cern.ch/PhysicsValidationDB/uploader

Prerequisits (MaCOSX)

Get homebrew form
http://brew.sh/
/usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"

brew install root6
brew info root6 → follow the instruction what to add to your .bashrc_profife
brew install homebrew/python/numpy

Then you should be ready to go:

python ./plot_histofiles.py --metadatafile meta_pion-_G4Al_totalxserr.json -c convert --output pion-_G4Al.json pion_xs.root:pion-_G4Al_totalxserr

Upload Tool: usage

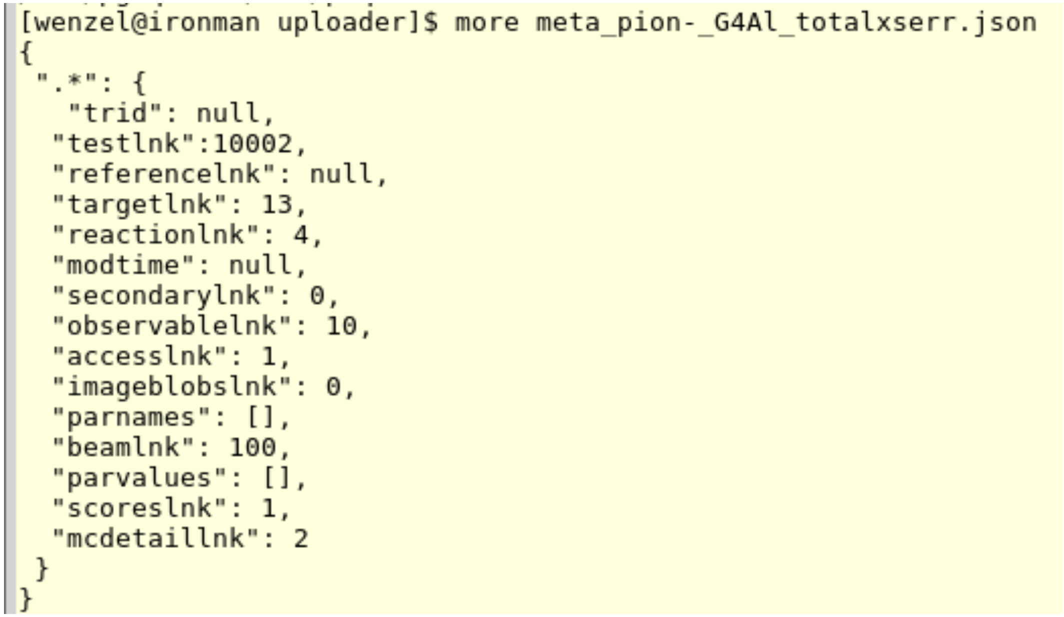
```
Usage:  plot_histofiles.py  [--comand|-c <cmd>] [--output|-o <ofile>] [--metadata|-m k[:type]=v] [--metadatafile <mdf>]
<files>
where:
  <files> are the files to read.
       File extension determines format. CSV is the text format
       from G4Analyais. For ROOT format, you need t specify the
       name of the file to be read in. Ex: file.root:h1
       pickle format is supported (file should be created with command "save")
  <cmd> is one of ("plot", "convert","save","genmd","list")
      "plot" (default) to plot the content of the file
      (requires matplotlib)
      "convert" creates an output file in JSON format suitable
      for FNALdb upload
      "save" saves histograms in internal format to pickle file "histos.pkl"
      "genmd" generates a metadata skeleton file as specified in <files>
      "list" shows content of ROOT File (TKey). Only for ROOT format.
  <ofile> is the output file name (default="output.json") for
      converted output for FNALdb
 <hn:k[:type]=v> is a key-value pair to add as metadata to FNALdb output
      hn is a regexp to assing the metadata to histogram based on names.
      k is the key of the metadata, type (default INT) can be INT if the value has to
      interpreted as integer value of STR if it must be interpreted as string
      or FLT if it is a floating point value
      e.g. -m .*A:INT=1 means add to all objects the integer metadata 1 with key A
 <mdf> is a json of pickle file containing the metadatada in a format of the type:
       { "regexpName" : { metadata } } where regexpName is a regular expression that
       matches a converted object name (the name being the ROOT TObject name or CSV full-filename)
```

Extracting TGraph from a root file and creating json file for upload to DoSSiER

Command used to add one record to the database
(cross sections for pion- on Aluminium)

```
python  ./plot_histofiles.py --metadatafile meta_pion-_G4Al_totalxserr.json \
-c convert  \
--output pion-_G4Al.json pion_xs.root:pion-_G4Al_totalxserr
```

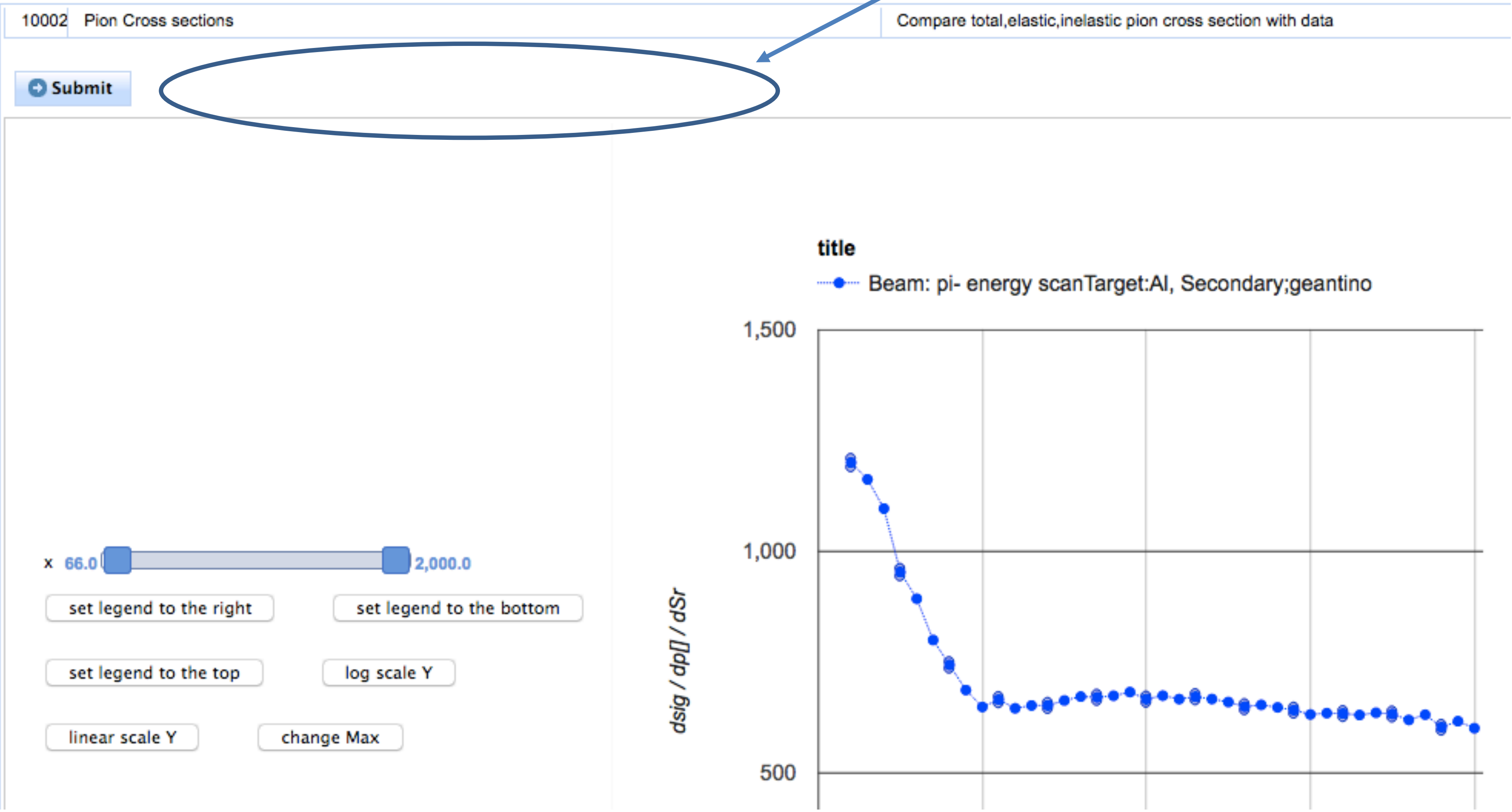
The Meta Data file: (in addition from the meta data provided by root)



```
[wenzel@ironman uploader]$ more meta_pion-_G4Al_totalxserr.json
{
 ".*": {
   "trid": null,
  "testlnk":10002,
  "referencelnk": null,
  "targetlnk": 13,
  "reactionlnk": 4,
  "modtime": null,
  "secondarylnk": 0,
  "observablelnk": 10,
  "accesslnk": 1,
  "imageblobslnk": 0,
  "parnames": [],
  "beamlnk": 100,
  "parvalues": [],
  "scoreslnk": 1,
  "mcdetaillnk": 2
 }
}
```

Data Base we start with

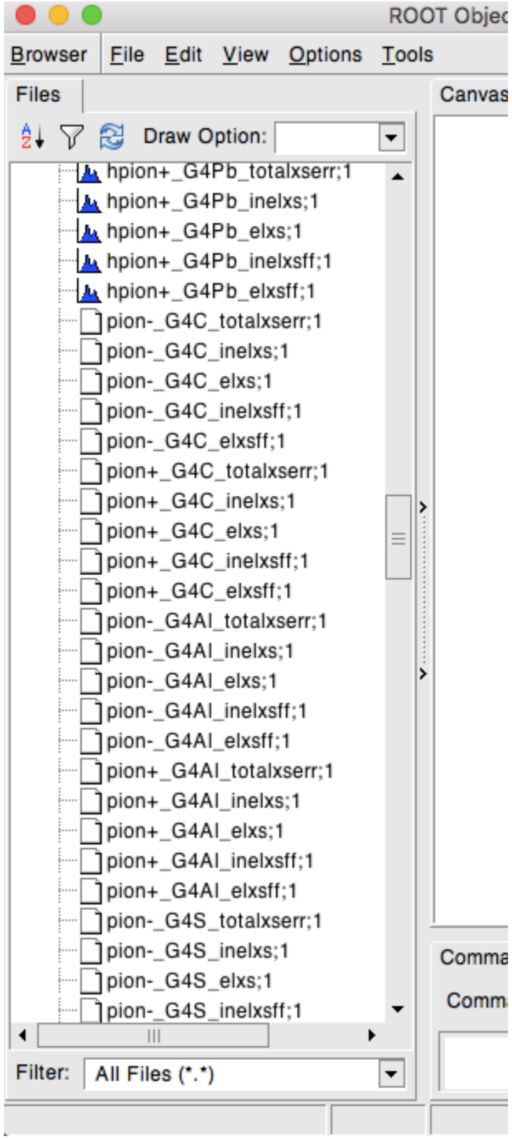
No selection Menus????



10002 Pion Cross sections

Compare total,elastic,inelastic pion cross section with data

root pion_xs.root

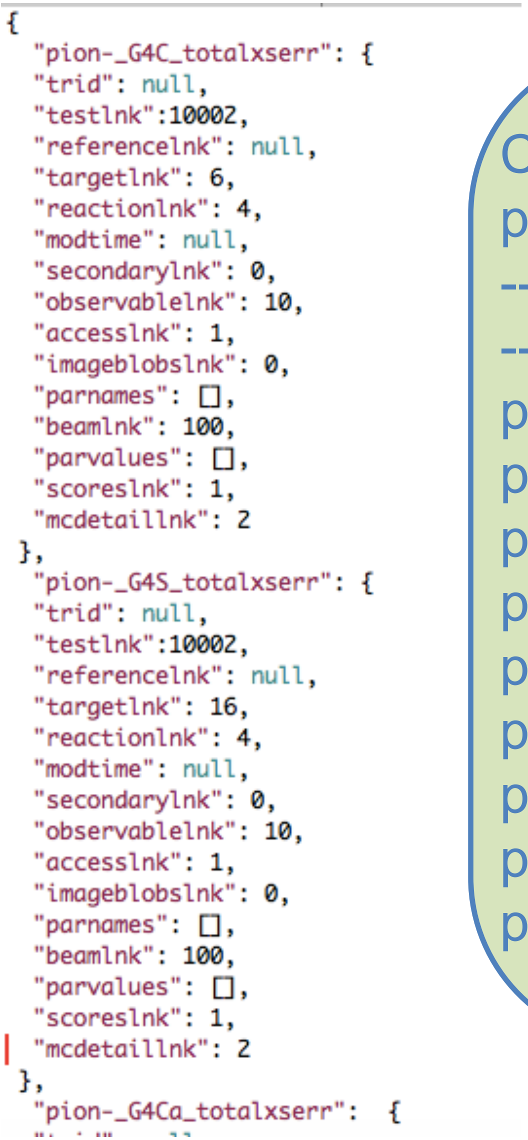


```
hpion+_G4Pb_totalxserr;1
hpion+_G4Pb_inelxs;1
hpion+_G4Pb_elxs;1
hpion+_G4Pb_inelxsff;1
hpion+_G4Pb_elxsff;1
pion-_G4C_totalxserr;1
pion-_G4C_inelxs;1
pion-_G4C_elxs;1
pion-_G4C_inelxsff;1
pion-_G4C_elxsff;1
pion+_G4C_totalxserr;1
pion+_G4C_inelxs;1
pion+_G4C_elxs;1
pion+_G4C_inelxsff;1
pion+_G4C_elxsff;1
pion-_G4Al_totalxserr;1
pion-_G4Al_inelxs;1
pion-_G4Al_elxs;1
pion-_G4Al_inelxsff;1
pion-_G4Al_elxsff;1
pion+_G4Al_totalxserr;1
pion+_G4Al_inelxs;1
pion+_G4Al_elxs;1
pion+_G4Al_inelxsff;1
pion+_G4Al_elxsff;1
pion-_G4S_totalxserr;1
pion-_G4S_inelxs;1
pion-_G4S_elxs;1
pion-_G4S_inelxsff;1
```

File contains pi+/pi- cross sections (total, elastic, inelastic) for various elements (C,Al,S,Ca,Fe,Cu,Ag,Sn,Au,Pb) both as histograms and TGraphError.

So first let’s enter the missing 9 entries for the pi- beam to the database

File with metadata: meta_pi-_all_totalxserr.json

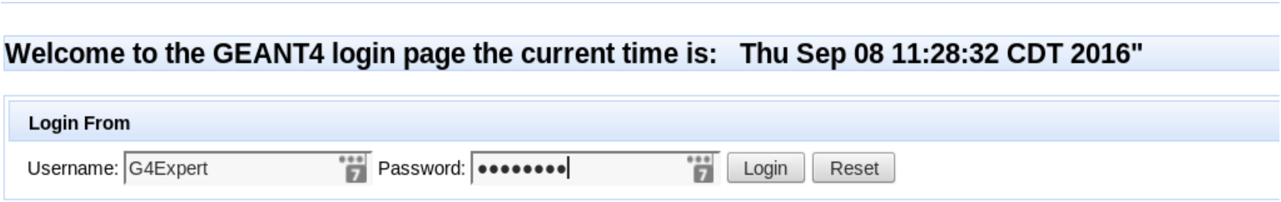


```
{
  "pion-_G4C_totalxserr": {
  "trid": null,
  "testlnk":10002,
  "referencelnk": null,
  "targetlnk": 6,
  "reactionlnk": 4,
  "modtime": null,
  "secondarylnk": 0,
  "observablelnk": 10,
  "accesslnk": 1,
  "imageblobslnk": 0,
  "parnames": □,
  "beamlnk": 100,
  "parvalues": □,
  "scoreslnk": 1,
  "mcdetaillnk": 2
 },
  "pion-_G4S_totalxserr": {
  "trid": null,
  "testlnk":10002,
  "referencelnk": null,
  "targetlnk": 16,
  "reactionlnk": 4,
  "modtime": null,
  "secondarylnk": 0,
  "observablelnk": 10,
  "accesslnk": 1,
  "imageblobslnk": 0,
  "parnames": □,
  "beamlnk": 100,
  "parvalues": □,
  "scoreslnk": 1,
  "mcdetaillnk": 2
 },
  "pion-_G4Ca_totalxserr":  {
```

Command to convert:

```
python  ./plot_histofiles.py \
--metadatafile meta_pi-_all_totalxserr.json -c convert  \
--output pi-_all.json \
pion_xs.root:pion-_G4C_totalxserr  \
pion_xs.root:pion-_G4S_totalxserr \
pion_xs.root:pion-_G4Ca_totalxserr \
pion_xs.root:pion-_G4Fe_totalxserr \
pion_xs.root:pion-_G4Cu_totalxserr \
pion_xs.root:pion-_G4Ag_totalxserr \
pion_xs.root:pion-_G4Sn_totalxserr \
pion_xs.root:pion-_G4Au_totalxserr \
pion_xs.root:pion-_G4Pb_totalxserr
```

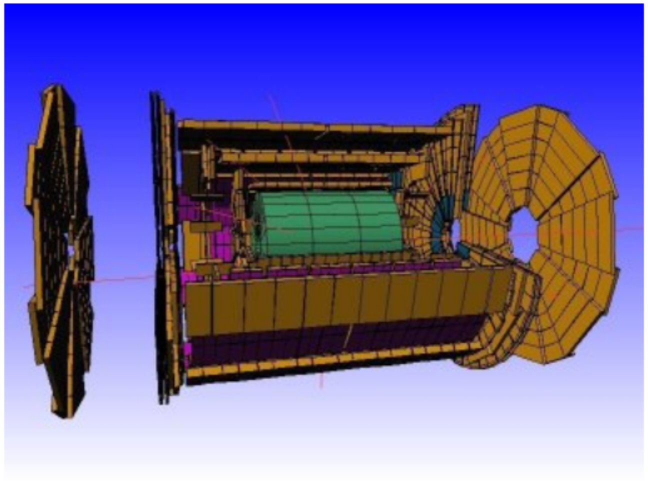
Uploading to DoSSiER: Log in to the Geant 4 Collaborators page



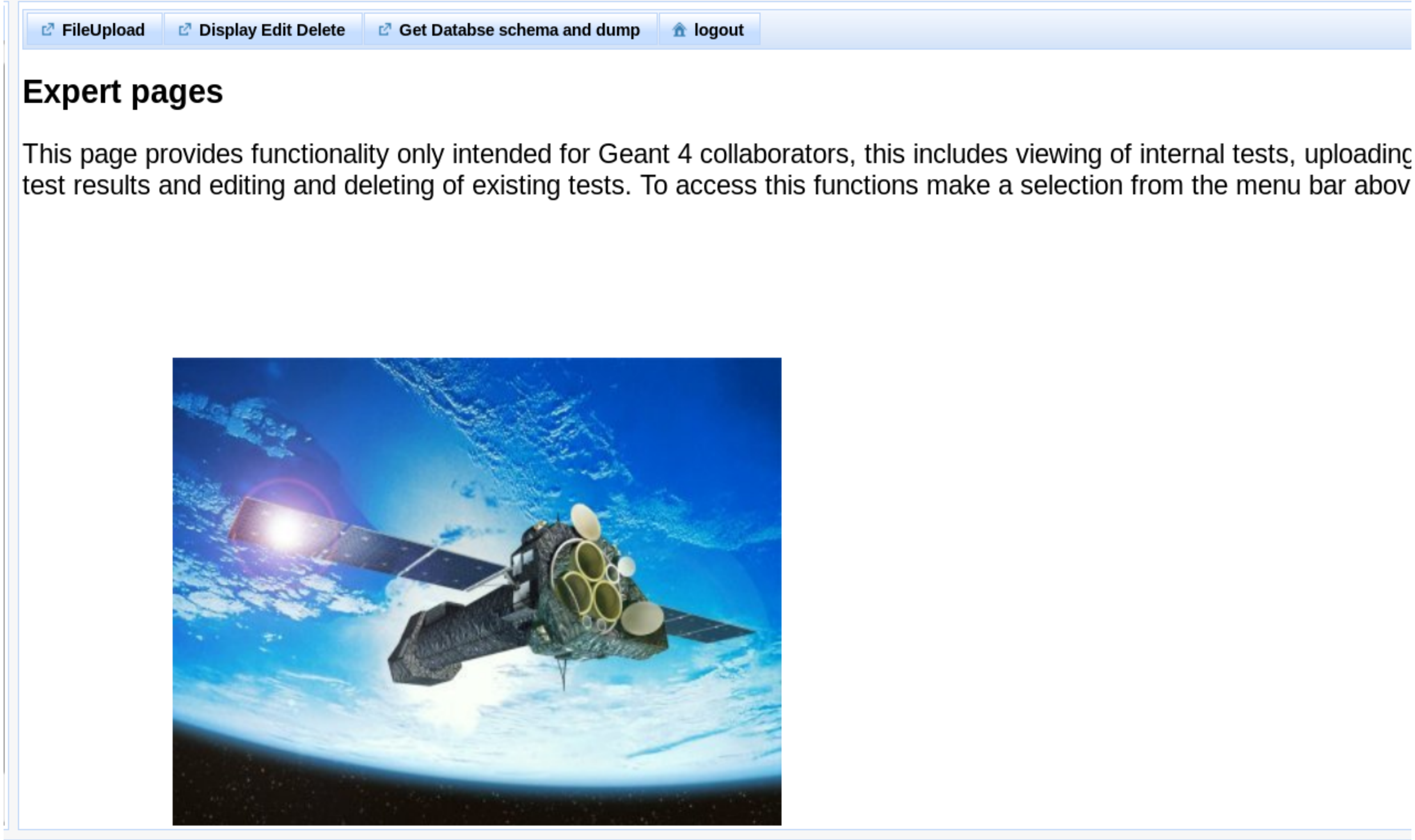
Welcome to the GEANT4 login page the current time is: Thu Sep 08 11:28:32 CDT 2016"

Login From

Username: G4Expert Password: •••••••••



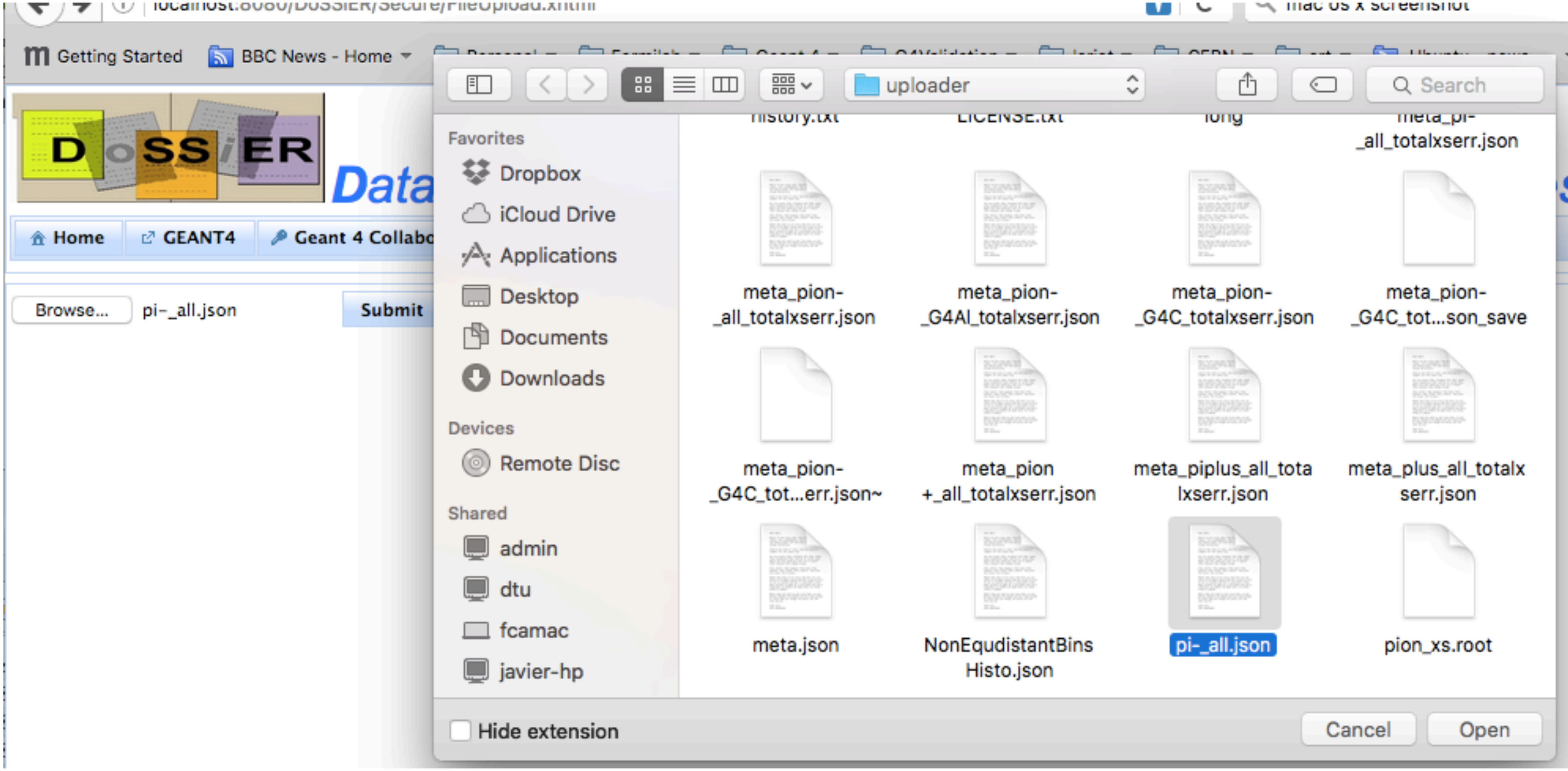
Uploading to DoSSiER: Now select FileUpload



Expert pages

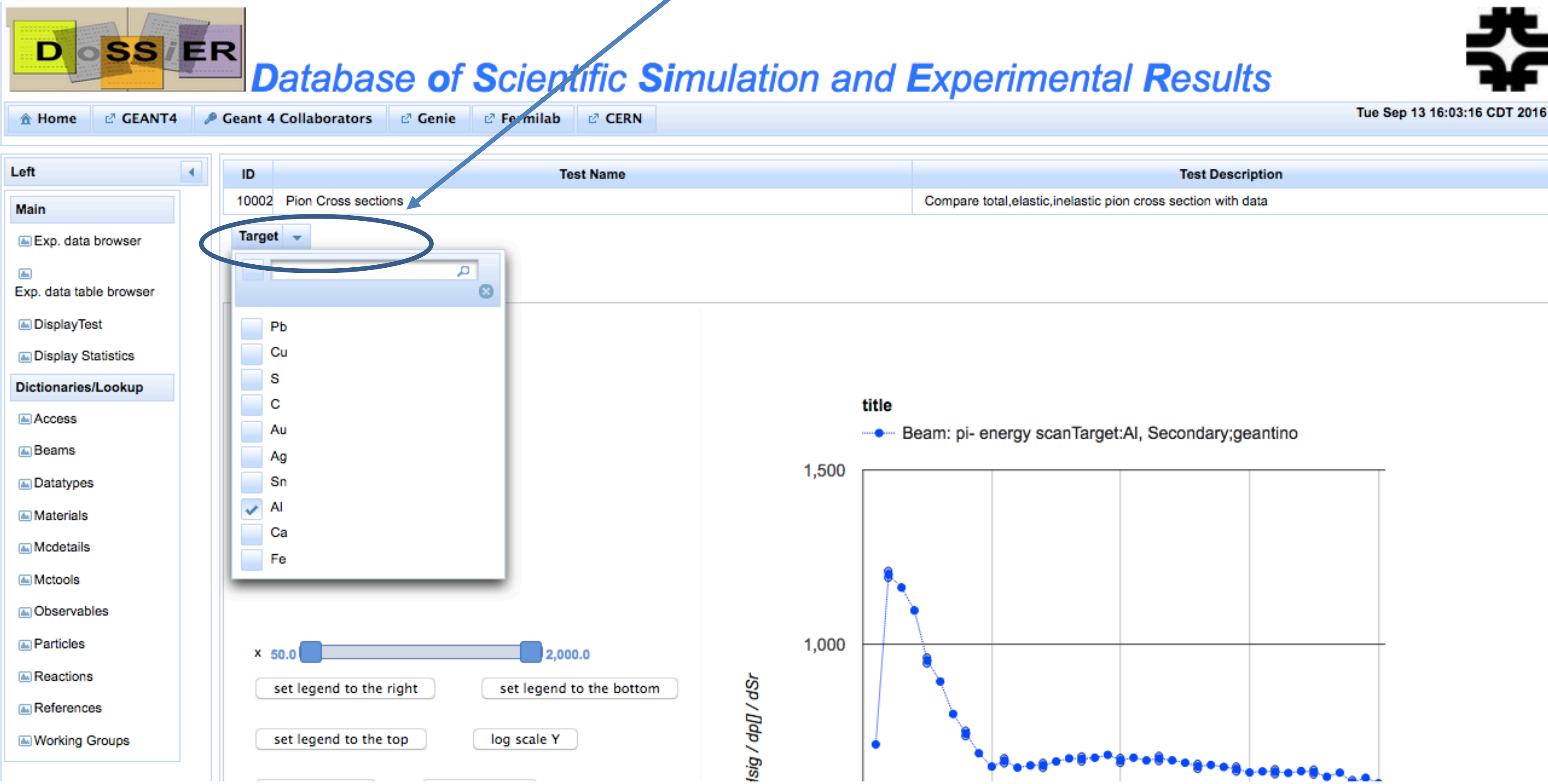
This page provides functionality only intended for Geant 4 collaborators, this includes viewing of internal tests, uploading test results and editing and deleting of existing tests. To access this functions make a selection from the menu bar abov

Uploading to DoSSiER: select the file to upload and press submit.



localhost:8080/DoSSiER/secure/FileUpload.xhtml
mac os x screenshot
Getting Started
BBC News - Home
DoSSiER
Data
Home
GEANT4
Geant 4 Collabo
Browse...
pi-_all.json
Submit
uploader
Search
Favorites
Dropbox
iCloud Drive
Applications
Desktop
Documents
Downloads
Devices
Remote Disc
Shared
admin
dtu
fcamac
javier-hp
history.txt
LICENSE.txt
long
meta_pi-_all_totalxserr.json
meta_pion-_all_totalxserr.json
meta_pion-_G4Al_totalxserr.json
meta_pion-_G4C_totalxserr.json
meta_pion-_G4C_tot...son_save
meta_pion-_G4C_tot...err.json~
meta_pion+_all_totalxserr.json
meta_piplus_all_totalxserr.json
meta_plus_all_totalxserr.json
meta.json
NonEqudistantBinsHisto.json
pi-_all.json
pion_xs.root
Hide extension
Cancel
Open

We got a Menu now!!!!!
Let’s select all

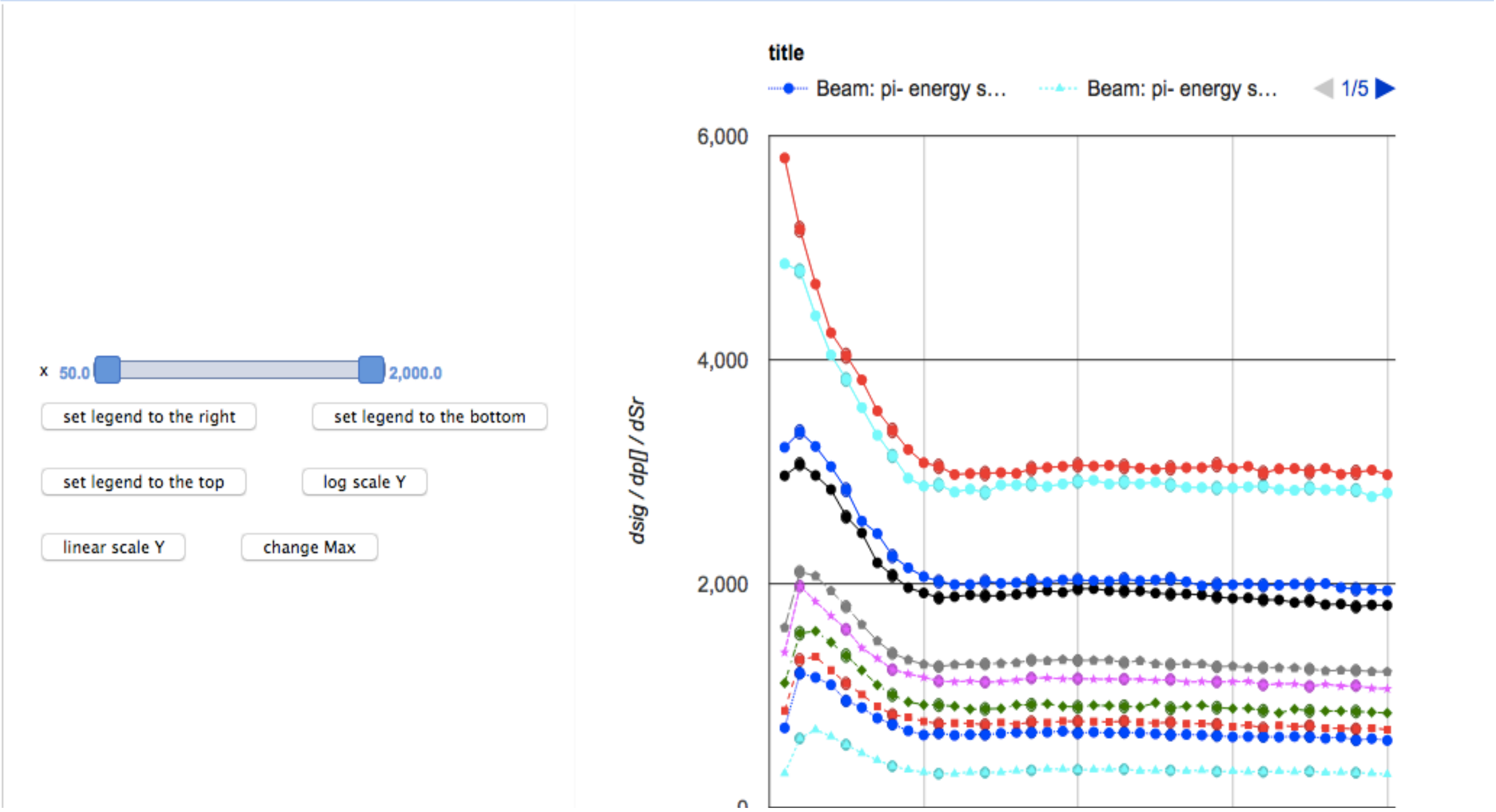


ID	Test Name	Test Description
10002	Pion Cross sections	Compare total,elastic,inelastic pion cross section with data

Target

- □ Pb
- □ Cu
- □ S
- □ C
- □ Au
- □ Ag
- □ Sn
- ☑ Al
- □ Ca
- □ Fe

Cool!!!!: pi- cross section for 10 different elements



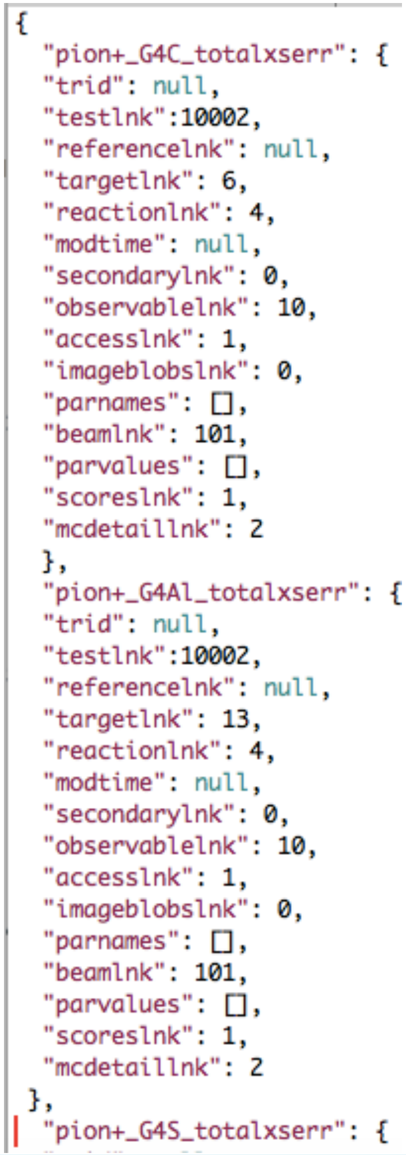
Conclusion

This wasn’t so difficult → No excuse not to try!!!!
Will add upload via web service soon → staged.
Move to fermilab/CERN SSO (saml/shibboleth)

Thanks to Andrea and Julia !!!!

Backup

Ok let’s also add the pi+



```
{
  "pion+_G4C_totalxserr": {
  "trid": null,
  "testlnk":10002,
  "referencelnk": null,
  "targetlnk": 6,
  "reactionlnk": 4,
  "modtime": null,
  "secondarylnk": 0,
  "observablelnk": 10,
  "accesslnk": 1,
  "imageblobslnk": 0,
  "parnames": [],
  "beamlnk": 101,
  "parvalues": [],
  "scoreslnk": 1,
  "mcdetaillnk": 2
  },
  "pion+_G4Al_totalxserr": {
  "trid": null,
  "testlnk":10002,
  "referencelnk": null,
  "targetlnk": 13,
  "reactionlnk": 4,
  "modtime": null,
  "secondarylnk": 0,
  "observablelnk": 10,
  "accesslnk": 1,
  "imageblobslnk": 0,
  "parnames": [],
  "beamlnk": 101,
  "parvalues": [],
  "scoreslnk": 1,
  "mcdetaillnk": 2
 },
  "pion+_G4S_totalxserr": {
```

Command to convert:

```
python  ./plot_histofiles.py \
--metadatafile meta_pi+_all_totalxserr.json -c convert  \
--output pi+_all.json \
pion_xs.root:pion+_G4C_totalxserr  \
pion_xs.root:pion+_G4Al_totalxserr \
pion_xs.root:pion+_G4S_totalxserr \
pion_xs.root:pion+_G4Ca_totalxserr \
pion_xs.root:pion+_G4Fe_totalxserr \
pion_xs.root:pion+_G4Cu_totalxserr \
pion_xs.root:pion+_G4Ag_totalxserr \
pion_xs.root:pion+_G4Sn_totalxserr \
pion_xs.root:pion+_G4Au_totalxserr \
pion_xs.root:pion+_G4Pb_totalxserr
```